```
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
```

```
**FILE**ID**IMGDECODE                                    G  3


      IIIIII   MM        MM  GGGGGGGG  DDDDDDDD    EEEEEEEEEE    CCCCCCCC    OOOOOO    DDDDDDDD    EEEEEEEEEE
      IIIIII   MM        MM  GGGGGGGG  DDDDDDDD    EEEEEEEEEE    CCCCCCCC    OOOOOO    DDDDDDDD    EEEEEEEEEE
        II     MMMM    MMMM  GG        DD      DD  EE          CC          OO      OO  DD      DD  EE
        II     MMMM    MMMM  GG        DD      DD  EE          CC          OO      OO  DD      DD  EE
        II     MM  MM  MM    GG        DD      DD  EE          CC          OO      OO  DD      DD  EE
        II     MM  MM  MM    GG        DD      DD  EE          CC          OO      OO  DD      DD  EE
        II     MM        MM  GG        DD      DD  EEEEEEEE    CC          OO      OO  DD      DD  EEEEEEEE
        II     MM        MM  GG        DD      DD  EEEEEEEE    CC          OO      OO  DD      DD  EEEEEEEE
        II     MM        MM  GG  GGGGGG  DD    DD  EE          CC          OO      OO  DD      DD  EE
        II     MM        MM  GG  GGGGGG  DD    DD  EE          CC          OO      OO  DD      DD  EE
        II     MM        MM  GG      GG  DD    DD  EE          CC          OO      OO  DD      DD  EE
        II     MM        MM  GG      GG  DD    DD  EE          CC          OO      OO  DD      DD  EE          ....
      IIIIII   MM        MM    GGGGGG  DDDDDDDD    EEEEEEEEEE    CCCCCCCC    OOOOOO    DDDDDDDD    EEEEEEEEEE  ....
      IIIIII   MM        MM    GGGGGG  DDDDDDDD    EEEEEEEEEE    CCCCCCCC    OOOOOO    DDDDDDDD    EEEEEEEEEE  ....
                                                                                                               ....

    LL          IIIIII      SSSSSSSS
    LL          IIIIII      SSSSSSSS
    LL            II      SS
    LL            II      SS
    LL            II      SS
    LL            II      SS
    LL            II        SSSSSS
    LL            II        SSSSSS
    LL            II              SS
    LL            II              SS
    LL            II              SS
    LL            II              SS
    LLLLLLLLLL  IIIIII    SSSSSSSS
    LLLLLLLLLL  IIIIII    SSSSSSSS
```

```
0 %TITLE  'Get and Decode Image Header and Sections'
0 MODULE IMG$DECODE (
0                 LANGUAGE (BLISS32),
0                 IDENT = 'V04-000'
0                 ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:     Exec, Shareable routines to decode image header and sections
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains the routines to retrieve and decode
1 !       an image header and the image section descriptors.
1 !
1 ! ENVIRONMENT:  VAX/VMS Operating System
1 !
1 ! AUTHOR:       Bob Grosso,     CREATION DATE:  16-Mar-1983
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-010 MSH0051         Michael S. Harvey       20-May-1984
1 !               Convert old format image name string to new format.
1 !
1 !       V03-009 MSH0043         Michael S. Harvey       8-May-1984
1 !               When converting x-linker image headers into a modern
1 !               form, update the image IDs correspondingly.
1 !
1 !       V03-008 MSH0041         Michael S. Harvey       2-May-1984
1 !               Add some beef to the bounds checking code to ensure that
1 !               only valid images are run. These checks filter obviously
1 !               bad image headers and images with bad ISD lists.
```

```
 1 !
 1 !        V03-007 LJK0269          Lawrence J. Kenah          31-Mar-1984
 1 !                Miscellaneous cleanup.
 1 !                Do not perform consistence checks on TYPE 2 images. They are
 1 !                not necessarily produced by the linker.
 1 !                Make sure that a primitive length check is performed on the
 1 !                IHD and ISD sizes before the buffer is copied.
 1 !
 1 !        V03-006 LJA0110          Laurie J. Anderson          6-Feb-1984
 1 !                Change the error messages returned from the image decode
 1 !                routines to be something more intelligent than "bad hdr".
 1 !
 1 !        V03-005 WMC0001          Wayne Cardoza              24-Jan-1984
 1 !                Add support for cross-linker and V3 FT1 images.
 1 !
 1 !        V03-004 LJK0243          Lawrence J. Kenah          23-Aug-1983
 1 !                Return IHD$Q_PRIVREQS of all privileges for old images,
 1 !                ones that do not contain a SYSVER field.
 1 !
 1 !        V03-003 LJK0234          Lawrence J. Kenah          26-Jul-1983
 1 !                Fix code that transforms old image header into latest
 1 !                form of image header.
 1 !
 1 !        V03-002 LJK0229          Lawrence J. Kenah          12-Jul-1983
 1 !                Treat the alias and offset as words. Treat the ISD
 1 !                size as a signed word.
 1 !
 1 !        V03-001 LJK0223          Lawrence J. Kenah           6-Jul-1983
 1 !                Make IHD and ISD sizes into words so that the comparisons
 1 !                are made correctly.
 1 !--
```

```
%SBTTL  'Definitions'

!
! INCLUDE FILES:
!

LIBRARY 'SYS$LIBRARY:LIB.L32';                          ! Define system data structures

REQUIRE 'LIB$:IMGMSGDEF.R32';                           ! Get status code definitions

!
! PSECT DECLARATIONS:
!

PSECT
        CODE    = YF$$SYSIMGACT (WRITE),
        PLIT    = YF$$SYSIMGACT (WRITE, EXECUTE);

!
! LITERALS
!

LITERAL
        TRUE = 1,
        FALSE = 0,

        IMG$C_BLOCKSIZ = 512;

!
! EXTERNAL REFERENCES:
!

EXTERNAL LITERAL

        EXE$C_SYSEFN : UNSIGNED (6);    ! System event flag for QIO Wait read

!
! FORWARD ROUTINE REFERENCES
!

FORWARD ROUTINE
        CONVERT_XLINK;

!
! Define VMS block structures
!
STRUCTURE
        BBLOCK [O, P, S, E; N] =
                [N]
                (BBLOCK + O) <P, S, E>;
```

IMG$
V04-

; Ro

; S

```
1 %SBTTL  'IMG$DECODE_IHD  Get Image Header'
1 GLOBAL ROUTINE IMG$DECODE_IHD
1         ( CHAN, BLK_BUFADR, IHD_BUFADR, VBN_ADR, OFFSET_ADR, HDRVER_ADR, ALIAS_ADR ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       Chan            Channel on which image file is open
1 !       Blk_bufadr      Address of buffer to contain 1st block of image
1 !       Ihd_bufadr      Address of buffer to contain decoded IHD
1 !       VBN_adr         Address of VBN to be set to 1
1 !       Offset_adr      Address of Offset in which to return offset to 1st ISD
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 LITERAL
2     IHDMAXSIZ = IHD$C_LENGTH +  ! Maximum length for fixed portion of header
2                 IHA$C_LENGTH +
2                 IHS$C_LENGTH +
2                 IHI$C_LENGTH +
2                 IHP$C_LENGTH ,
2     IHI_S_IMGNAM = 16;          ! Length of image name string prior to VMS V4
2
2 LOCAL
2     B_IHD : REF BBLOCK,         ! Buffer IHD
2     IHD : REF BBLOCK,
2     IOSB : BBLOCK [8],          ! Quadword IO status block
2     HDR_INSERT,
2     IHI_INSERT,
2     OFF2 : WORD,
2     SIZE,
2     STATUS;                     ! Status
2
2 BIND
2     V4_MAJORID = UPLIT (%ASCII'02'),          ! Major ID for VMS V4 images
2     V4_MINORID = UPLIT (%ASCII'05'),          ! Minor ID for VMS V4 images
2     HEADER_VERSION = .HDRVER_ADR            : WORD,
2     LAST_WORD = .ALIAS_ADR                  : SIGNED WORD,
2     OFFSET = .OFFSET_ADR                    : WORD,
2     VBN = .VBN_ADR;
```

```
                                                              L 3
IMG$DECODE          Get and Decode Image Header and Sections        16-Sep-1984 02:41:10    VAX-11 Bliss-32 V4.0-742           Page   5
V04-000             IMG$DECODE_IHD  Get Image Header                14-Sep-1984 13:12:35    [SYS.SRC]IMGDECODE.B32;1                 (3)

;   199          0282  2
;   200          0283  2  VBN = 1;                          ! Read from first block
;   201          0284  2  SIZE = IMG$C_BLOCKSIZ;            ! Read one block
;   202          0285  2
;   203          0286  2
;   204          0287  2  !
;   205          0288  2  !      Read first block
;   206          0289  2  !
;   207        P 0290  2  STATUS = $QIOW (
;   208        P 0291  2                      EFN = EXE$C_SYSEFN,      ! Event flag
;   209        P 0292  2                      CHAN = .CHAN,            ! Channel
;   210        P 0293  2                      FUNC = IO$_READVBLK,     ! Read a virtual block
;   211        P 0294  2                      IOSB = IOSB,             ! I/O status block
;   212        P 0295  2                      P1 = .BLK_BUFADR,        ! Buffer to read in to
;   213        P 0296  2                      P2 = .SIZE,              ! Number of bytes to read
;   214        P 0297  2                      P3 = .VBN                ! Virtual block number to read
;   215          0298  2                     );
;   216          0299  2
;   217          0300  2  IF .STATUS
;   218          0301  2  THEN
;   219          0302  2      STATUS = .IOSB [0,0,16,0];            ! Pick up final status
;   220          0303  2  IF NOT .STATUS
;   221          0304  2  THEN
;   222          0305  2      RETURN .STATUS;
;   223          0306  2
;   224          0307  2  B_IHD = .BLK_BUFADR;                      ! Image header
;   225          0308  2  LAST_WORD = .B_IHD [IHD$W_ALIAS];         ! Contents of last word of header block
;   226          0309  2
;   227          0310  2  !
;   228          0311  2  !      Process the image based upon which type of image it is. Screen
;   229          0312  2  !      out obvious image pretenders.
;   230          0313  2  !
;   231          0314  2  CASE .LAST_WORD
;   232          0315  2  FROM IHD$C_MINCODE TO IHD$C_MAXCODE OF
;   233          0316  2      SET
;   234          0317  2
;   235          0318  2      [IHD$C_RSX, IHD$C_BPA, IHD$C_ALIAS] :
;   236          0319  2
;   237          0320  3          BEGIN
;   238          0321  3          CH$MOVE (IMG$C_BLOCKSIZ, .B_IHD, .IHD_BUFADR);  ! Copy image header to buffer
;   239          0322  3          HEADER_VERSION = 0;
;   240          0323  3          RETURN SS$_NORMAL;
;   241          0324  2          END;
;   242          0325  2
;   243          0326  2      [IHD$C_NATIVE, IHD$C_CLI] :
;   244          0327  2
;   245          0328  3          BEGIN
;   246          0329  3          IF .B_IHD [IHD$W_MAJORID] EQL IHX$K_MAJORID     ! If Cross linker format
;   247          0330  3          THEN
;   248          0331  4              BEGIN
;   249          0332  4              HEADER_VERSION = IHD$C_GEN_XLNKR;
;   250          0333  4              STATUS = CONVERT_XLINK (.BLK_BUFADR, .IHD_BUFADR);
;   251          0334  4              OFFSET = .B_IHD [IHD$W_SIZE];
;   252          0335  4              RETURN .STATUS;
;   253          0336  3              END;
;   254          0337  3
;   255          0338  3          !
```

```
	! Check for a reasonable header record size and set of offsets.
	! Simply verify that the offsets and the regions they point to
	! fall within the image header record.
	!
	OFFSET = .B_IHD [IHD$W_SIZE];
	IF (.OFFSET LSSU $BYTEOFFSET(IHD$L_LNKFLAGS))
	    OR
	    (.OFFSET GTRU IHDMAXSIZ)
	THEN
	    RETURN IMG$_IMG_SIZ;

	!	Verify range of activation data offset

	OFF2 = .B_IHD [IHD$W_ACTIVOFF];
	IF (.OFF2 LSSU $BYTEOFFSET(IHD$L_LNKFLAGS))
	    OR
	    (.OFF2 + IHA$C_LENGTH GTRU IHDMAXSIZ)
	THEN
	    RETURN IMG$_BADOFFSET;

	!	Verify range of debug and global symbol table offset

	IF .B_IHD [IHD$W_SYMDBGOFF] NEQ 0
	THEN
	    BEGIN
	    OFF2 = .B_IHD [IHD$W_SYMDBGOFF];
	    IF (.OFF2 LSSU $BYTEOFFSET(IHD$L_LNKFLAGS))
		OR
		(.OFF2 + IHS$C_LENGTH GTRU IHDMAXSIZ)
	    THEN
		RETURN IMG$_BADOFFSET;
	    END;

	!	Verify range of image ID data offset

	OFF2 = .B_IHD [IHD$W_IMGIDOFF];
	IF (.OFF2 LSSU $BYTEOFFSET(IHD$L_LNKFLAGS))
	    OR
	    (.OFF2 + IHI$C_LENGTH GTRU IHDMAXSIZ)
	THEN
	    RETURN IMG$_BADOFFSET;

	!	Verify range of patch data offset

	IF .B_IHD [IHD$W_PATCHOFF] NEQ 0
	THEN
	    BEGIN
	    OFF2 = .B_IHD [IHD$W_PATCHOFF];
	    IF (.OFF2 LSSU $BYTEOFFSET(IHD$L_LNKFLAGS))
		OR
		(.OFF2 + IHP$C_LENGTH GTRU IHDMAXSIZ)
	    THEN
		RETURN IMG$_BADOFFSET;
	    END;

	!
	! Copy image header to buffer
```

```
3         !
3         CH$MOVE (.B_IHD [IHD$W_SIZE], .B_IHD, .IHD_BUFADR);
3
3         HDR_INSERT = 0;                               ! Length by which header will be pried open
3         HEADER_VERSION = IHD$C_GEN_FIXUP;             ! Default to most current
3         IHD = .IHD_BUFADR;
3
3         !
3         ! Calculate the degree by which the fixed portion of this header
3         ! differs from the current format of the fixed part of an image header.
3         ! Then, expand the fixed portion of the header by the required amount,
3         ! thus converting it to the current format as if the image had been
3         ! relinked.
3         !
3         IF $BYTEOFFSET (IHD$L_LNKFLAGS) GEQ .IHD [IHD$W_ACTIVOFF]
3         THEN                                          ! Link flags were not present
4             BEGIN                                     ! so insert a longword
4             HDR_INSERT = 4;
4             HEADER_VERSION = IHD$C_GEN_NATIVE;
3             END;
3
3         IF $BYTEOFFSET (IHD$L_SYSVER) GEQ .IHD [IHD$W_ACTIVOFF]
3         THEN                                          ! System version and Ident were not pressent
4             BEGIN                                     ! so insert two blank longwords
4
4             BIND
4                 PRIVILEGE_MASK = IHD [IHD$Q_PRIVREQS]    : VECTOR [2];
4
4             HDR_INSERT = .HDR_INSERT + 8;
4             HEADER_VERSION = IHD$C_GEN_LNKFLG;
4             PRIVILEGE_MASK [0] = -1;                  ! Insure that image privilege mask
4             PRIVILEGE_MASK [1] = -1;                  !  indicates that all privileges are set
3             END;
3
3         IF $BYTEOFFSET (IHD$L_IAFVA) GEQ .IHD [IHD$W_ACTIVOFF]
3         THEN                                          ! Relative virtual address of fixup vector
4             BEGIN                                     ! not present so insert a blank longword
4             HDR_INSERT = .HDR_INSERT + 4;
4             HEADER_VERSION = IHD$C_GEN_SYSVER;
3             END;
3
3
3         IF .HDR_INSERT NEQ 0
3         THEN                                          ! Shift non-fixed portion of image
4             BEGIN                                     ! to insert missing part of fixed section
4
4             CH$MOVE (
4                 (.IHD [IHD$W_SIZE] - .IHD [IHD$W_ACTIVOFF]),      ! Shift the portion beginning at the
4                 (.IHD + .IHD [IHD$W_ACTIVOFF]),                   !  point located by the first offset
4                 (.IHD + .IHD [IHD$W_ACTIVOFF] + .HDR_INSERT));    !  by the amount to be inserted
4
4             CH$FILL (0, .HDR_INSERT,                  ! Fill the space created for the insert
4                 .IHD + .IHD [IHD$W_ACTIVOFF]);
3             END;
3
3         !
3         ! Determine the extent that the image ident area differs in size from
```

**FI

```
	    ! the current format. Expand the image ident area by the required
	    ! amount, thus converting to the current format without relinking.
	    !
	    IHI_INSERT = 0;                         ! Assume no conversion required
	    IF (.IHD [IHD$W_MAJORID] LSSU .V4_MAJORID)
	        OR
	        (
	         (.IHD [IHD$W_MAJORID] EQL .V4_MAJORID)
	         AND
	         (.IHD [IHD$W_MINORID] LSSU .V4_MINORID)
	        )
	    THEN
	        !
	        ! The image name string grew between VMS V3 and V4. Split the
	        ! image ident area after the old image name string and expand
	        ! the string to the current maximum size, zero filled.
	        !
	        BEGIN
	        IHI_INSERT = IHI$S_IMGNAM - IHI_S_IMGNAM; ! Calculate size difference
	        CH$MOVE (
	            (.IHD [IHD$W_SIZE] - (.IHD [IHD$W_IMGIDOFF] + IHI_S_IMGNAM)),
	            (.IHD + .IHD [IHD$W_IMGIDOFF] + IHI_S_IMGNAM),
	            (.IHD + .IHD [IHD$W_IMGIDOFF] + IHI$S_IMGNAM) );
	        CH$FILL (0, .IHI_INSERT,
	            (.IHD + .IHD [IHD$W_IMGIDOFF] + IHI_S_IMGNAM) );
	        END;
	!
	! Correct all the offsets to compensate for the insertion(s). Note that two of
	! the offsets locate optional parts of the image header and are only updated
	! if the associated areas are present in the image (offsets are nonzero).
	!
	    IF (.HDR_INSERT NEQ 0)
	        OR
	        (.IHI_INSERT NEQ 0)
	    THEN
	        BEGIN
	        IHD [IHD$W_SIZE]     = .IHD [IHD$W_SIZE]     + .HDR_INSERT + .IHI_INSERT;
	        IHD [IHD$W_ACTIVOFF] = .IHD [IHD$W_ACTIVOFF] + .HDR_INSERT;
	        IHD [IHD$W_IMGIDOFF] = .IHD [IHD$W_IMGIDOFF] + .HDR_INSERT;

	        IF .IHD [IHD$W_SYMDBGOFF] NEQU 0
	        THEN
	            IHD [IHD$W_SYMDBGOFF] = .IHD [IHD$W_SYMDBGOFF] + .HDR_INSERT;

	        IF .IHD [IHD$W_PATCHOFF] NEQU 0
	        THEN
	            IHD [IHD$W_PATCHOFF]  = .IHD [IHD$W_PATCHOFF]  + .HDR_INSERT + .IHI_INSERT;

	        END;

	    RETURN SS$_NORMAL;
	    END;

	[INRANGE,OUTRANGE] :

	    RETURN IMG$_BADHDR;                ! Unrecognizable or unsupported image type
```

```
;  427        0510  2       TES;                                    ! CASE of image types
;  428        0511  2
;  429        0512  1 END;                                          ! IMG$DECODE_IHD routine

                                                                            .TITLE  IMG$DECODE Get and Decode Image Header and Sect
;                                                                                               ions
                                                                            .IDENT  \V04-000\

                                                                            .PSECT  YF$$SYSIMGACT,2

                                       00  00  32  30  00000 P.AAA:   .ASCII  \02\<0><0>                                    ;
                                       00  00  35  30  00004 P.AAB:   .ASCII  \05\<0><0>                                    ;

                                                             V4_MAJORID=           P.AAA
                                                             V4_MINORID=           P.AAB
                                                                            .EXTRN  EXE$C_SYSEFN, SYS$QIOW

                                                  07FC 00000                .ENTRY  IMG$DECODE_IHD, Save R2,R3,R4,R5,R6,R7,R8,- ; 0226
                                                                                    R9,R10                                  ;
                              5E                08  C2 00002                SUBL2   #8, SP                                  ;
                              58          18    AC  D0 00005                MOVL    HDRVER_ADR, R8                          ; 0278
                        10    BC                01  D0 00009                MOVL    #1, @VBN_ADR                            ; 0283
                              50        0200    8F  3C 0000D                MOVZWL  #512, SIZE                              ; 0284
                                                7E  7C 00012                CLRQ    -(SP)                                   ; 0298
                                                7E  D4 00014                CLRL    -(SP)                                   ;
                                          10    BC  DD 00016                PUSHL   @VBN_ADR                                ;
                                                50  DD 00019                PUSHL   SIZE                                    ;
                                          08    AC  DD 0001B                PUSHL   BLK_BUFADR                              ;
                                                7E  7C 0001E                CLRQ    -(SP)                                   ;
                                          20    AE  9F 00020                PUSHAB  IOSB                                    ;
                                                31  DD 00023                PUSHL   #49                                     ;
                                          04    AC  DD 00025                PUSHL   CHAN                                    ;
                              7E               00G  9A 00028                MOVZBL  S^EXE$C_SYSEFN, -(SP)                   ;
                   00000000G  00                0C  FB 0002B                CALLS   #12, SYS$QIOW                           ;
                              57                50  D0 00032                MOVL    R0, STATUS                              ;
                              50                57  E9 00035                BLBC    STATUS, 4$                              ; 0300
                              57                6E  3C 00038                MOVZWL  IOSB, STATUS                            ; 0302
                              4A                57  E9 0003B                BLBC    STATUS, 4$                              ; 0303
                              56          08    AC  D0 0003E                MOVL    BLK_BUFADR, B_IHD                       ; 0307
                        1C    BC        01FE    C6  B0 00042                MOVW    510(B_IHD), @ALIAS_ADR                  ; 0308
                04    FFFF    8F          1C    BC  AF 00048                CASEW   @ALIAS_ADR, #-1, #4                     ; 0328
      0012          0012            0012      001E     0004F 1$:    .WORD   3$-1$,-                                 ;
                                              001E     00057                        2$-1$,-                                 ;
                                                                                    2$-1$,-                                 ;
                                                                                    2$-1$,-                                 ;
                                                                                    3$-1$                                   ;
                              50 084D8C84       8F  D0 00059                MOVL    #139299972, R0                          ; 0508
                                                    04 00060                RET                                             ;
              0C     BC       66        0200    8F  28 00061 2$:    MOVC3   #512, (B_IHD), @IHD_BUFADR              ; 0321
                                                68  B4 00068                CLRW    (R8)                                    ; 0322
                                              017A  31 0006A                BRW     19$                                     ; 0328
                      3130    8F          0C    A6  B1 0006D 3$:    CMPW    12(B_IHD), #12592                       ; 0329
                                                17  12 00073                BNEQ    5$                                      ;
                              68                01  B0 00075                MOVW    #1, (R8)                                ; 0332
                              7E          08    AC  7D 00078                MOVQ    BLK_BUFADR, -(SP)                       ; 0333
                      0000V   CF                02  FB 0007C                CALLS   #2, CONVERT_XLINK                       ;
```

```
                          57                  50  D0 00081         MOVL    R0, STATUS                                   :
                  14      BC                  66  B0 00084         MOVW    (B_IHD), @OFFSET_ADR                         : 0334
                          50                  57  D0 00088 4$:     MOVL    STATUS, R0                                   : 0335
                                                  04 0008B         RET
                  14      BC                  66  B0 0008C 5$:     MOVW    (B_IHD), @OFFSET_ADR                         : 0343
                          20          14      BC  B1 00090         CMPW    @OFFSET_ADR, #32                             : 0344
                                              08  1F 00094         BLSSU   6$
                00D4      8F          14      BC  B1 00096         CMPW    @OFFSET_ADR, #212                            : 0346
                                              08  1B 0009C         BLEQU   7$
                       50 084D8CA4            8F  D0 0009E 6$:     MOVL    #139300004, R0                               : 0348
                                                  04 000A5         RET
                          50          02      A6  B0 000A6 7$:     MOVW    2(B_IHD), OFF2                               : 0352
                          20                  50  B1 000AA         CMPW    OFF2, #32                                    : 0353
                                              62  1F 000AD         BLSSU   9$
                          51                  50  3C 000AF         MOVZWL  OFF2, R1                                     : 0355
                          51                  14  C0 000B2         ADDL2   #20, R1
            000000D4      8F                  51  D1 000B5         CMPL    R1, #212
                                              53  1A 000BC         BGTRU   9$
                                      04      A6  B5 000BE         TSTW    4(B_IHD)                                     : 0361
                                              18  13 000C1         BEQL    8$
                          50          04      A6  B0 000C3         MOVW    4(B_IHD), OFF2                               : 0364
                          20                  50  B1 000C7         CMPW    OFF2, #32                                    : 0365
                                              45  1F 000CA         BLSSU   9$
                          51                  50  3C 000CC         MOVZWL  OFF2, R1                                     : 0367
                          51                  14  C0 000CF         ADDL2   #20, R1
            000000D4      8F                  51  D1 000D2         CMPL    R1, #212
                                              36  1A 000D9         BGTRU   9$
                          50          06      A6  B0 000DB 8$:     MOVW    6(B_IHD), OFF2                               : 0374
                          20                  50  B1 000DF         CMPW    OFF2, #32                                    : 0375
                                              2D  1F 000E2         BLSSU   9$
                          51                  50  3C 000E4         MOVZWL  OFF2, R1                                     : 0377
                          51          50      A1  9E 000E7         MOVAB   80(R1), R1
            000000D4      8F                  51  D1 000EB         CMPL    R1, #212
                                              1D  1A 000F2         BGTRU   9$
                                      08      A6  B5 000F4         TSTW    8(B_IHD)                                     : 0383
                                              20  13 000F7         BEQL    10$
                          50          08      A6  B0 000F9         MOVW    8(B_IHD), OFF2                               : 0386
                          20                  50  B1 000FD         CMPW    OFF2, #32                                    : 0387
                                              0F  1F 00100         BLSSU   9$
                          50                  50  3C 00102         MOVZWL  OFF2, R0                                     : 0389
                          50                  2C  C0 00105         ADDL2   #44, R0
            000000D4      8F                  50  D1 00108         CMPL    R0, #212
                                              08  1B 0010F         BLEQU   10$
                       50 084D8C94            8F  D0 00111 9$:     MOVL    #139299988, R0                               : 0391
                                                  04 00118         RET
      0C    BC            66                  66  28 00119 10$:    MOVC3   (B_IHD), (B_IHD), @IHD_BUFADR                : 0397
                                              59  D4 0011E         CLRL    HDR_INSERT                                   : 0399
                          68                  05  B0 00120         MOVW    #5, (R8)                                     : 0400
                          56          0C      AC  D0 00123         MOVL    IHD_BUFADR, IHD                              : 0401
                          20          02      A6  B1 00127         CMPW    2(IHD), #32                                  : 0410
                                              06  1A 0012B         BGTRU   11$
                          59                  04  D0 0012D         MOVL    #4, HDR_INSERT                               : 0413
                          68                  02  B0 00130         MOVW    #2, (R8)                                     : 0414
                          28          02      A6  B1 00133 11$:    CMPW    2(IHD), #40                                  : 0417
                                              11  1A 00137         BGTRU   12$
                          50          14      A6  9E 00139         MOVAB   20(IHD), R0                                  : 0422
                          59                  08  C0 0013D         ADDL2   #8, HDR_INSERT                               : 0424
```

```
IMG$
V04-
;  1
;  1
;  1
;  1
;  1
;  1
;  1
;  1
;  1
;  1
;  1
;  1
;  1
;  1
```

```
                            68          03  B0 00140        MOVW    #3, (R8)                                       ; 0425
                            60          01  CE 00143        MNEGL   #1, (R0)                                       ; 0426
                    04      A0          01  CE 00146        MNEGL   #1, 4(R0)                                      ; 0427
                            50      02  A6  3C 0014A 12$:   MOVZWL  2(IHD), R0                                     ; 0430
                            2C          50  B1 0014E        CMPW    R0, #44
                                        06  1A 00151        BGTRU   13$
                            59          04  C0 00153        ADDL2   #4, HDR_INSERT                                 ; 0433
                            68          04  B0 00156        MOVW    #4, (R8)                                       ; 0434
                                        5A  D4 00159 13$:   CLRL    R10                                            ; 0438
                                        59  D5 0015B        TSTL    HDR_INSERT
                                        17  13 0015D        BEQL    14$
                                        5A  D6 0015F        INCL    R10
                            51          66  3C 00161        MOVZWL  (IHD), R1                                      ; 0443
                            51          50  C2 00164        SUBL2   R0, R1
                57          56          50  C1 00167        ADDL3   R0, IHD, R7                                    ; 0444
              6947          67          51  28 0016B        MOVC3   R1, (R7), (HDR_INSERT)[R7]                     ; 0445
     59         00          6E          00  2C 00170        MOVC5   #0, (SP), #0, HDR_INSERT, (R7)                 ; 0448
                                        67     00175
                            58          D4 00176 14$:       CLRL    IHI_INSERT                                     ; 0456
   FE78  CF     0C  A6      10          00  ED 00178        CMPZV   #0, #16, 12(IHD), V4_MAJORID                   ; 0457
                                        14  1F 00180        BLSSU   15$
   FE6E  CF     0C  A6      10          00  ED 00182        CMPZV   #0, #16, 12(IHD), V4_MAJORID                   ; 0460
                                        2B  12 0018A        BNEQ    16$
   FE68  CF     0E  A6      10          00  ED 0018C        CMPZV   #0, #16, 14(IHD), V4_MINORID                   ; 0462
                                        21  1E 00194        BGEQU   16$
                            58          18  D0 00196 15$:   MOVL    #24, IHI_INSERT                                ; 0471
                            50      06  A6  3C 00199        MOVZWL  6(IHD), R0                                     ; 0473
                            51          66  3C 0019D        MOVZWL  (IHD), R1
                            51          50  C2 001A0        SUBL2   R0, R1
                            51          10  C2 001A3        SUBL2   #16, R1
                57          56          50  C1 001A6        ADDL3   R0, IHD, R7                                    ; 0474
            28  A7      10  A7          51  28 001AA        MOVC3   R1, 16(R7), 40(R7)                             ; 0475
     58         00          6E          00  2C 001B0        MOVC5   #0, (SP), #0, IHI_INSERT, 16(R7)               ; 0477
                                    10  A7     001B5
                            04          5A  E8 001B7 16$:   BLBS    R10, 17$                                       ; 0484
                                        58  D5 001BA        TSTL    IHI_INSERT                                     ; 0486
                                        29  13 001BC        BEQL    19$
                            50          66  3C 001BE 17$:   MOVZWL  (IHD), R0                                      ; 0489
                            50          59  C0 001C1        ADDL2   HDR_INSERT, R0
                66          50          58  A1 001C4        ADDW3   IHI_INSERT, R0, (IHD)
                    02      A6          59  A0 001C8        ADDW2   HDR_INSERT, 2(IHD)                             ; 0490
                    06      A6          59  A0 001CC        ADDW2   HDR_INSERT, 6(IHD)                             ; 0491
                                    04  A6  B5 001D0        TSTW    4(IHD)                                         ; 0493
                                        04  13 001D3        BEQL    18$
                    04      A6          59  A0 001D5        ADDW2   HDR_INSERT, 4(IHD)                             ; 0495
                            50      08  A6  3C 001D9 18$:   MOVZWL  8(IHD), R0                                     ; 0497
                                        08  13 001DD        BEQL    19$
                            50          59  C0 001DF        ADDL2   HDR_INSERT, R0                                 ; 0499
            08  A6          50          58  A1 001E2        ADDW3   IHI_INSERT, R0, 8(IHD)
                            50          01  D0 001E7 19$:   MOVL    #1, R0                                         ; 0503
                                            04 001EA        RET                                                    ; 0512

; Routine Size:  491 bytes,    Routine Base:  YF$$SYSIMGACT + 0008


;  430          0513  1
```

```
1 %SBTTL  'IMG$GET_NEXT_ISD  Get Image Section Descriptor'
1 GLOBAL ROUTINE IMG$GET_NEXT_ISD
1    ( CHAN, BLK_BUFADR, IHD_BUFADR, VBN_ADR, OFFSET_ADR, ISD_BUFADR, HEADER_VERSION ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !        Chan            Channel on which image file is open
1 !        Blk_bufadr      Address of buffer which contains block of image header
1 !        Ihd_bufadr      Address of buffer containing decoded IHD
1 !        VBN_adr         Address of VBN in blk_bufadr
1 !        Offset_adr      Address of Offset to ISD
1 !        ISD_bufadr      Address of buffer to contain decoded ISD
1 !
1 ! IMPLICIT INPUTS:
1 !        NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !        NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !        NONE
1 !
1 ! SIDE EFFECTS:
1 !        NONE
1 !
1 !--
1
2 BEGIN
2 LOCAL
2     IOSB                    : BBLOCK [8],   ! Quadword IO status block
2     B_ISD                   : REF BBLOCK,   ! ISD is header block buffer
2     ISD                     : REF BBLOCK,
2     ISD_SIZ                 : SIGNED WORD,
2     SIZE,
2     STATUS;                                 ! Status
2
2 BIND
2     IHD = .IHD_BUFADR               : BBLOCK,
2     OFFSET = .OFFSET_ADR            : WORD,
2     VBN = .VBN_ADR;
2
2 !
2 !   Validate that offset and VBN are reasonable
2 !
2 IF .OFFSET GEQU
3        (IF .VBN EQL 1
3        THEN IMG$C_BLOCKSIZ - 2
3        ELSE IMG$C_BLOCKSIZ)
2 THEN
2     RETURN IMG$_ISD_OFF;
2
2 IF .VBN GTR .IHD [IHD$B_HDRBLKCNT]
2 THEN
```

```
  2         RETURN IMG$_ISD_VBN;
  2
  2     !
  2     !   Get next ISD
  2     !
  2     B_ISD = .BLK_BUFADR + .OFFSET;
  2     ISD_SIZ = .B_ISD [ISD_W_SIZE];
  2
  2     !
  2     !   See whether offset points off the block and we need to read the next block
  2     !
  2     IF .ISD_SIZ EQL -1
  2     THEN                                        ! Read next block
  3         BEGIN
  3         VBN = .VBN + 1;                         ! Increment VBN
  3         OFFSET = 0;
  3         SIZE = IMG$C_BLOCKSIZ;
  3
P 3         STATUS = $QIOW
P 3                     (
P 3                     EFN = EXE$C_SYSEFN,     ! Event flag
P 3                     CHAN = .CHAN,           ! Channel
P 3                     FUNC = IO$_READVBLK,    ! Read a virtual block
P 3                     IOSB = IOSB,            ! I/O status block
P 3                     P1 = .BLK_BUFADR,       ! Buffer to read in to
P 3                     P2 = .SIZE,             ! Number of bytes to read
P 3                     P3 = .VBN               ! Virtual block number to read
  3                     );
  3
  3         IF .STATUS
  3         THEN
  3             STATUS = .IOSB [0,0,16,0];          ! Pick up final status
  3         IF NOT .STATUS
  3         THEN
  3             RETURN .STATUS;
  3
  3         B_ISD = .BLK_BUFADR;
  3         ISD_SIZ = .B_ISD [ISD_W_SIZE];
  3
  3         IF .ISD_SIZ EQL -1                      ! Trap consecutive 'wrap' ISDs
  3         THEN
  3             RETURN IMG$_INCONISD;
  3
  2         END;
  2
  2     !
  2     !   See whether there are any ISDs left
  2     !
  2     IF .ISD_SIZ EQL 0
  2     THEN                                        ! No more ISDs left
  2         RETURN IMG$_ENDOFHDR;
  2
  2     !
  2     !   Validate that the ISD size is reasonable
  2     !
  3     IF (.ISD_SIZ LSS ISD$C_LENDZRO)
  2         OR
```

```
;  546        0628  3         (.ISD_SIZ GTR ISD$C_MAXLENGLBL)
;  547        0629  2     THEN
;  548        0630  2         RETURN IMG$_ISD_SIZ;
;  549        0631  2
;  550        0632  2     !
;  551        0633  2     !   Make sure that ISD doesn't attempt to wrap around to the next block
;  552        0634  2     !
;  553        0635  2     IF (.OFFSET + .ISD_SIZ) GTRU
;  554        0636  3             (IF .VBN EQL 1
;  555        0637  3             THEN IMG$C_BLOCKSIZ - 2
;  556        0638  3             ELSE IMG$C_BLOCKSIZ)
;  557        0639  2     THEN
;  558        0640  2         RETURN IMG$_INCONISD;
;  559        0641  2
;  560        0642  2     ISD = .ISD_BUFADR;
;  561        0643  2     CH$MOVE (.ISD_SIZ, .B_ISD, .ISD);         ! Copy from block to ISD buffer
;  562        0644  2     OFFSET = .OFFSET + .ISD_SIZ;
;  563        0645  2
;  564        0646  2     !
;  565        0647  2     ! Don't use page fault cluster size for cross-linker images
;  566        0648  2     !
;  567        0649  2     IF .HEADER_VERSION EQL IHD$C_GEN_XLNKR
;  568        0650  2     THEN
;  569        0651  2         ISD [ISD$B_PFC] = 0;
;  570        0652  2
;  571        0653  2     !
;  572        0654  2     ! Some V3 images use IHD$L_IAFVA to identify the fixup vectors
;  573        0655  2     !
;  574        0656  2     IF .HEADER_VERSION EQL IHD$C_GEN_FIXUP
;  575        0657  2     THEN
;  576        0658  2         IF (.ISD [ISD$V_VPN] * 512) EQL .IHD [IHD$L_IAFVA]
;  577        0659  2             AND
;  578        0660  2             .ISD [ISD$V_VPN] NEQ 0
;  579        0661  2         THEN
;  580        0662  2             ISD [ISD$V_FIXUPVEC] = 1;
;  581        0663  2
;  582        0664  2     RETURN SS$_NORMAL;
;  583        0665  1     END;                                      ! IMG$GET_NEXT_ISD routine




                                                03FC 00000           .ENTRY  IMG$GET_NEXT_ISD, Save R2,R3,R4,R5,R6,R7,-   ; 0515
                                                                             R8,R9
                                       5E         08   C2 00002      SUBL2   #8, SP                                        ;
                                       59    0C   AC   D0 00005      MOVL    IHD_BUFADR, R9                                ; 0555
                                       58    14   AC   D0 00009      MOVL    OFFSET_ADR, R8                                ; 0556
                                       52    10   AC   D0 0000D      MOVL    VBN_ADR, R2                                   ; 0557
                                       01         62   D1 00011      CMPL    (R2), #1                                      ; 0563
                                                  07   12 00014      BNEQ    1$                                            ;
                                       50  01FE   8F   3C 00016      MOVZWL  #510, R0                                      ; 0564
                                                  05   11 0001B      BRB     2$                                            ;
                                       50  0200   8F   3C 0001D 1$:  MOVZWL  #512, R0                                      ; 0563
             50           68           10         00   ED 00022 2$:  CMPZV   #0, #16, (R8), R0                             ;
                                                  08   1F 00027      BLSSU   3$                                            ;
                                       50 084D8CB4 8F  D0 00029      MOVL    #139300020, R0                                ; 0567
```

```
                                                              I  4
IMG$DECODE        Get and Decode Image Header and Sections    16-Sep-1984 02:41:10    VAX-11 Bliss-32 V4.0-742          Page  15
V04-000           IMG$GET_NEXT_ISD  Get Image Section Descriptor 14-Sep-1984 13:12:35 [SYS.SRC]IMGDECODE.B32;1                (4)

                                              04 00030        RET                                          ;
      62      10   A9           08         00 ED 00031 3$:    CMPZV   #0, #8, 16(R9), (R2)                 ; 0569
                                           08 18 00037        BGEQ    4$                                   ;
                            50 084D8CBC    8F D0 00039        MOVL    #139300028, R0                       ; 0571
                                              04 00040        RET                                          ;
                            53             68 3C 00041 4$:    MOVZWL  (R8), B_ISD                          ; 0576
                            53         08  AC C0 00044        ADDL2   BLK_BUFADR, B_ISD                    ;
                            57             63 B0 00048        MOVW    (B_ISD), ISD_SIZ                     ; 0577
                  FFFF      8F             57 B1 0004B        CMPW    ISD_SIZ, #-1                         ; 0582
                                           40 12 00050        BNEQ    7$                                   ;
                                           62 D6 00052        INCL    (R2)                                 ; 0585
                                           68 B4 00054        CLRW    (R8)                                 ; 0586
                            50       0200  8F 3C 00056        MOVZWL  #512, SIZE                           ; 0587
                                           7E 7C 0005B        CLRQ    -(SP)                                ; 0598
                                           7E D4 0005D        CLRL    -(SP)                                ;
                                           62 DD 0005F        PUSHL   (R2)                                 ;
                                           50 DD 00061        PUSHL   SIZE                                 ;
                                       08  AC DD 00063        PUSHL   BLK_BUFADR                           ;
                                           7E 7C 00066        CLRQ    -(SP)                                ;
                                       20  AE 9F 00068        PUSHAB  IOSB                                 ;
                                           31 DD 0006B        PUSHL   #49                                  ;
                                       04  AC DD 0006D        PUSHL   CHAN                                 ;
                            7E            00G 9A 00070        MOVZBL  S^EXE$C_SYSEFN, -(SP)                ;
                 00000000G  00             0C FB 00073        CALLS   #12, SYS$QIOW                        ;
                            03             50 E9 0007A        BLBC    STATUS, 5$                           ; 0600
                            50             6E 3C 0007D        MOVZWL  IOSB, STATUS                         ; 0602
                            01             50 E8 00080 5$:    BLBS    STATUS, 6$                           ; 0603
                                              04 00083        RET                                          ;
                            53         08  AC D0 00084 6$:    MOVL    BLK_BUFADR, B_ISD                    ; 0607
                            57             63 B0 00088        MOVW    (B_ISD), ISD_SIZ                     ; 0608
                  FFFF      8F             57 B1 0008B        CMPW    ISD_SIZ, #-1                         ; 0610
                                           3F 13 00090        BEQL    13$                                  ;
                                           57 B5 00092 7$:    TSTW    ISD_SIZ                              ; 0619
                                           08 12 00094        BNEQ    8$                                   ;
                            50 084D8640    8F D0 00096        MOVL    #139298368, R0                       ; 0621
                                              04 0009D        RET                                          ;
                            0C             57 B1 0009E 8$:    CMPW    ISD_SIZ, #12                         ; 0626
                                           07 19 000A1        BLSS    9$                                   ;
                  0040      8F             57 B1 000A3        CMPW    ISD_SIZ, #64                         ; 0628
                                           08 15 000A8        BLEQ    10$                                  ;
                            50 084D8CC4    8F D0 000AA 9$:    MOVL    #139300036, R0                       ; 0630
                                              04 000B1        RET                                          ;
                            51             68 3C 000B2 10$:   MOVZWL  (R8), R1                             ; 0635
                            50             57 32 000B5        CVTWL   ISD_SIZ, R0                          ;
                            51             50 C0 000B8        ADDL2   R0, R1                               ;
                            01             62 D1 000BB        CMPL    (R2), #1                             ; 0636
                                           07 12 000BE        BNEQ    11$                                  ;
                            50       01FE  8F 3C 000C0        MOVZWL  #510, R0                             ; 0637
                                           05 11 000C5        BRB     12$                                  ;
                            50       0200  8F 3C 000C7 11$:   MOVZWL  #512, R0                             ; 0636
                            50             51 D1 000CC 12$:   CMPL    R1, R0                               ;
                                           08 1B 000CF        BLEQU   14$                                  ;
                            50 084D8CAC    8F D0 000D1 13$:   MOVL    #139300012, R0                       ; 0640
                                              04 000D8        RET                                          ;
                            56         18  AC D0 000D9 14$:   MOVL    ISD_BUFADR, ISD                      ; 0642
              66            63             57 28 000DD        MOVC3   ISD_SIZ, (B_ISD), (ISD)              ; 0643
                            68             57 A0 000E1        ADDW2   ISD_SIZ, (R8)                        ; 0644
```

```
                             01         1C   AC  D1 000E4         CMPL    HEADER_VERSION, #1                       : 0649
                                             03  12 000E8         BNEQ    15$                                      :
                                        07   A6  94 000EA         CLRB    7(ISD)                                   : 0651
                             05         1C   AC  D1 000ED  15$:   CMPL    HEADER_VERSION, #5                       : 0656
                                             1C  12 000F1         BNEQ    16$                                      :
         50       04   A6    15              00  EF 000F3         EXTZV   #0, #21, 4(ISD), R0                      : 0658
                       50    50              09  78 000F9         ASHL    #9, R0, R0                               :
                        2C   A9              50  D1 000FD         CMPL    R0, 44(R9)                               :
                                             0C  12 00101         BNEQ    16$                                      :
         00       04   A6    15              00  ED 00103         CMPZV   #0, #21, 4(ISD), #0                      : 0660
                                             04  13 00109         BEQL    16$                                      :
                        09   A6              04  88 0010B         BISB2   #4, 9(ISD)                               : 0662
                             50              01  D0 0010F  16$:   MOVL    #1, R0                                   : 0664
                                                 04 00112         RET                                              : 0665

; Routine Size:  275 bytes,    Routine Base:  YF$$SYSIMGACT + 01F3


;  584          0666  1
```

IMG$
V04-

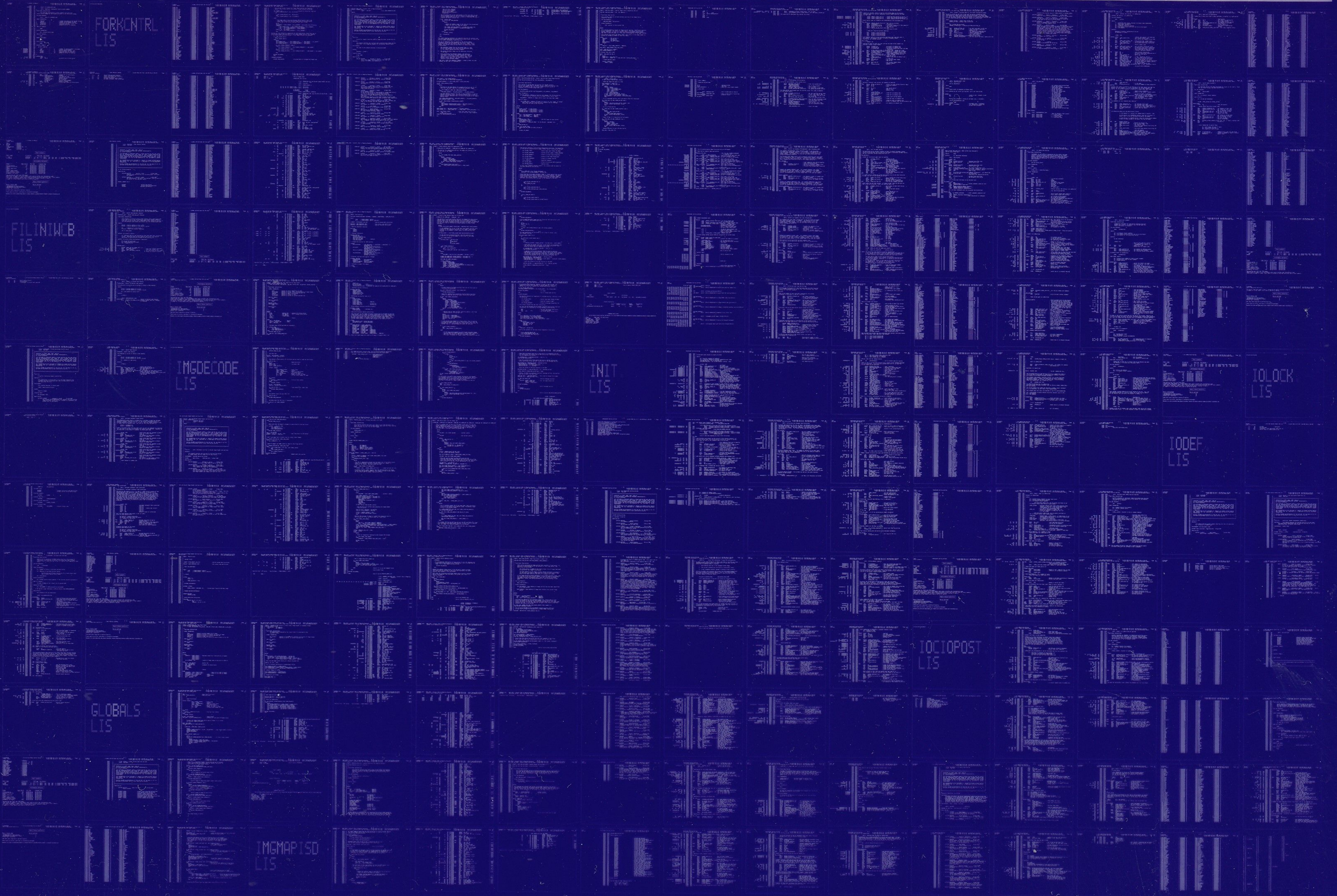
```
1 %SBTTL  'CONVERT_XLINK Convert a cross-linker image header to standard format'
1 ROUTINE CONVERT_XLINK
1    ( BLK_BUFADR   : REF $BBLOCK,
1      IHD          : REF $BBLOCK ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !         An image header produced by the cross-linker is converted to the
1 !         standard format.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !         Blk_bufadr          Address of buffer which contains first block of image header
1 !         Ihd                 Address of buffer to contain decoded IHD
1 !
1 ! IMPLICIT INPUTS:
1 !         NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !         NONE
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !         NONE
1 !
1 ! SIDE EFFECTS:
1 !         NONE
1 !
1 !--
1
2 BEGIN
2
2 BIND
2     PRIV_MASK = IHD [IHD$Q_PRIVREQS]   : VECTOR [2],
2     IHD_ACT_ADR = .IHD + IHD$K_LENGTH : VECTOR [3],
2     IHX_ACT_ADR = BLK_BUFADR [IHX$Q_STARTADR] : VECTOR [2],
2     IHS         = .IHD + IHD$K_LENGTH + IHA$K_LENGTH : $BBLOCK;
2 !
2 ! Zero the one page buffer which will contain decoded IHD
2 !
2 CH$FILL (0, 512, .IHD);
2 !
2 ! Fill in offsets and directly transportable fields
2 !
2 IHD [IHD$W_ACTIVOFF] = IHD$K_LENGTH;
2 IHD [IHD$W_SIZE] = IHD$K_LENGTH + IHA$K_LENGTH + IHS$K_LENGTH;
2 IHD [IHD$B_HDRBLKCNT] = .BLK_BUFADR [IHX$B_HDRBLKCNT];
2 !
2 ! Convert image ID fields
2 !
2 IHD [IHD$W_MAJORID] = IHD$K_MAJORID;
2 IHD [IHD$W_MINORID] = IHD$K_MINORID;
2 !
2 ! Assume all privileges
2 !
2 PRIV_MASK [0]= -1;
2 PRIV_MASK [1]= -1;
```

```
;  643          0724  2 !
;  644          0725  2 ! Add image activation data
;  645          0726  2 !
;  646          0727  2 IHD_ACT_ADR [0] = .IHX_ACT_ADR [0];
;  647          0728  2 IHD_ACT_ADR [1] = .IHX_ACT_ADR [1];
;  648          0729  2 !
;  649          0730  2 ! Check for DEBUG data
;  650          0731  2 !
;  651          0732  2 IF .BLK_BUFADR [IHX$W_MINORID] GEQ IHX$K_MINORID1
;  652          0733  2 THEN
;  653          0734  3     BEGIN
;  654          0735  3     IHD [IHD$W_SYMDBGOFF] = IHD$K_LENGTH + IHA$K_LENGTH;
;  655          0736  3     IHD_ACT_ADR [2] = .BLK_BUFADR [IHX$L_TFRADR3];
;  656          0737  3     IHS [IHS$L_DSTVBN] = .BLK_BUFADR [IHX$L_DSTVBN] ;
;  657          0738  3     IHS [IHS$L_GSTVBN] = .BLK_BUFADR [IHX$L_GSTVBN] ;
;  658          0739  3     IHS [IHS$W_DSTBLKS] = .BLK_BUFADR [IHX$W_DSTBLKS] ;
;  659          0740  3     IHS [IHS$W_GSTRECS] = .BLK_BUFADR [IHX$W_GSTRECS] ;
;  660          0741  2     END;
;  661          0742  2
;  662          0743  2 RETURN SS$_NORMAL;
;  663          0744  1 END;



                                                    0FFC 00000 CONVERT_XLINK:
                                                                          .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 0668
                              56        08    AC   D0 00002               MOVL    IHD, R6                                       ; 0700
                              5B        14    A6   9E 00006               MOVAB   20(R6), R11                                   ;
                              58        30    A6   9E 0000A               MOVAB   48(R6), R8                                    ; 0701
                              57        04    AC   D0 0000E               MOVL    BLK_BUFADR, R7                                ; 0702
                              5A        04    A7   9E 00012               MOVAB   4(R7), R10                                    ;
                              59        44    A6   9E 00016               MOVAB   68(R6), R9                                    ; 0703
   0200    8F            00   6E              00   2C 0001A               MOVC5   #0, (SP), #0, #512, (R6)                      ; 0707
                                              66      00021                                                                 ;
                              66  00300058    8F   D0 00022               MOVL    #3145816, (R6)                                ; 0712
                         10   A6        02    A7   90 00029               MOVB    2(R7), 16(R6)                                 ; 0713
                         0C   A6  35303230    8F   D0 0002E               MOVL    #892351024, 12(R6)                            ; 0717
                              6B              01   CE 00036               MNEGL   #1, (R11)                                     ; 0722
                         04   AB              01   CE 00039               MNEGL   #1, 4(R11)                                    ; 0723
                              68              6A   7D 0003D               MOVQ    (R10), (R8)                                   ; 0727
                       3130   8F        0E    A7   B1 00040               CMPW    14(R7), #12592                                ; 0732
                                              13   1F 00046               BLSSU   1$                                            ;
                         04   A6        44    8F   9B 00048               MOVZBW  #68, 4(R6)                                    ; 0735
                         08   A8        34    A7   D0 0004D               MOVL    52(R7), 8(R8)                                 ; 0736
                              69        28    A7   7D 00052               MOVQ    40(R7), (R9)                                  ; 0737
                         08   A9        30    A7   D0 00056               MOVL    48(R7), 8(R9)                                 ; 0739
                              50              01   D0 0005B 1$:           MOVL    #1, R0                                        ; 0743
                                                   04 0005E               RET                                                   ; 0744

; Routine Size:  95 bytes,    Routine Base:  YF$$SYSIMGACT + 0306
```

```
                                                                           M  4
IMG$DECODE         Get and Decode Image Header and Sections           16-Sep-1984 02:41:10     VAX-11 Bliss-32 V4.0-742              Page  19
V04-000            CONVERT_XLINK Convert a cross-linker image head 14-Sep-1984 13:12:35     [SYS.SRC]IMGDECODE.B32;1                    (6)

;  665             0745  1 END                                   !End of module IMGDECODE
;  666             0746  0 ELUDOM


;                                    PSECT SUMMARY
;
;       Name                        Bytes                        Attributes
;
;   YF$$SYSIMGACT                         869  NOVEC,  WRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                            Library Statistics
;
;                                            --------- Symbols --------        Pages        Processing
;       File                                 Total     Loaded   Percent        Mapped       Time
;
;   _$255$DUA28:[SYSLIB]LIB.L32;1            18619        63         0         1000          00:01.8


;                                    COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:IMGDECODE/OBJ=OBJ$:IMGDECODE MSRC$:IMGDECODE/UPDATE=(ENH$:IMGDECODE)

; Size:            861 code + 8 data bytes
; Run Time:           00:19.1
; Elapsed Time:       00:22.0
; Lines/CPU Min:       2348
; Lexemes/CPU-Min: 16064
; Memory Used:  205 pages
; Compilation Complete
```

0375 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY


FORKCNTRL
LIS

FILINIWCB
LIS

IMGDECODE
LIS

INIT
LIS

IOLOCK
LIS

IODEF
LIS

IOCIOPOST
LIS

GLOBALS
LIS

IMGMAPISD
LIS